Network Working Group
Request for Comments: 2516
Category: Informational
L. Mamakos
K. Lidl
J. Evarts
UUNET Technologies, Inc.
D. Carrel
D. Simone
RedBack Networks, Inc.
R. Wheeler
RouterWare, Inc.
February 1999

A Method for Transmitting PPP Over Ethernet (PPPoE)

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

The Point-to-Point Protocol (PPP) [1] provides a standard method for transporting multi-protocol datagrams over point-to-point links.

This document describes how to build PPP sessions and encapsulate PPP packets over Ethernet.

Applicability

This specification is intended to provide the facilities which are defined for PPP, such as the Link Control Protocol, Network-layer Control Protocols, authentication, and more. These capabilities require a point-to-point relationship between the peers, and are not designed for the multi-point relationships which are available in Ethernet and other multi-access environments.

This specification can be used by multiple hosts on a shared, Ethernet to open PPP sessions to multiple destinations via one or more bridging modems. It is intended to be used with broadband remote access technologies that provide a bridged Ethernet topology, when access providers wish to maintain the session abstraction associated with PPP.

This document describes the PPP Over Ethernet encapsulation that is being deployed by RedBack Networks, RouterWare, UUNET and others.

1. Introduction

Modern access technologies are faced with several conflicting goals. It is desirable to connect multiple hosts at a remote site through the same customer premise access device. It is also a goal to provide access control and billing functionality in a manner similar to dial-up services using PPP. In many access technologies, the most cost effective method to attach multiple hosts to the customer premise access device, is via Ethernet. In addition, it is desirable to keep the cost of this device as low as possible while requiring little or no configuration.

PPP over Ethernet (PPPoE) provides the ability to connect a network of hosts over a simple bridging access device to a remote Access Concentrator. With this model, each host utilizes it’s own PPP stack and the user is presented with a familiar user interface. Access control, billing and type of service can be done on a per-user, rather than a per-site, basis.

To provide a point-to-point connection over Ethernet, each PPP session must learn the Ethernet address of the remote peer, as well as establish a unique session identifier. PPPoE includes a discovery protocol that provides this.

2. Conventions

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [2].

3. Protocol Overview

PPPoE has two distinct stages. There is a Discovery stage and a PPP Session stage. When a Host wishes to initiate a PPPoE session, it must first perform Discovery to identify the Ethernet MAC address of the peer and establish a PPPoE SESSION_ID. While PPP defines a peer-to-peer relationship, Discovery is inherently a client-server relationship. In the Discovery process, a Host (the client) discovers an Access Concentrator (the server). Based on the network topology, there may be more than one Access Concentrator that the Host can communicate with. The Discovery stage allows the Host to discover all Access Concentrators and then select one. When Discovery completes successfully, both the Host and the selected Access Concentrator have the information they will use to build their point-to-point connection over Ethernet.

The Discovery stage remains stateless until a PPP session is established. Once a PPP session is established, both the Host and the Access Concentrator MUST allocate the resources for a PPP virtual interface.

4. Payloads

The following packet formats are defined here. The payload contents will be defined in the Discovery and PPP sections.

An Ethernet frame is as follows:

```
                                     1
                 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                |       DESTINATION_ADDR        |
                |          (6 octets)           |
                |                               |
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                |         SOURCE_ADDR           |
                |          (6 octets)           |
                |                               |
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                |    ETHER_TYPE  (2 octets)     |
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                ~                               ~
                ~           payload             ~
                ~                               ~
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                |           CHECKSUM            |
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The DESTINATION_ADDR field contains either a unicast Ethernet destination address, or the Ethernet broadcast address (0xffffffff). For Discovery packets, the value is either a unicast or broadcast address as defined in the Discovery section. For PPP session traffic, this field MUST contain the peer's unicast address as determined from the Discovery stage.

The SOURCE_ADDR field MUST contains the Ethernet MAC address of the source device.

The ETHER_TYPE is set to either 0x8863 (Discovery Stage) or 0x8864 (PPP Session Stage).

The Ethernet payload for PPPoE is as follows:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  VER  | TYPE  |      CODE     |          SESSION_ID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LENGTH             |           payload             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The VER field is four bits and MUST be set to 0x1 for this version of the PPPoE specification.

The TYPE field is four bits and MUST be set to 0x1 for this version of the PPPoE specification.

The CODE field is eight bits and is defined below for the Discovery and PPP Session stages.

The SESSION_ID field is sixteen bits. It is an unsigned value in network byte order. It’s value is defined below for Discovery packets. The value is fixed for a given PPP session and, in fact, defines a PPP session along with the Ethernet SOURCE_ADDR and DESTINATION_ADDR. A value of 0xffff is reserved for future use and MUST NOT be used

The LENGTH field is sixteen bits. The value, in network byte order, indicates the length of the PPPoE payload. It does not include the length of the Ethernet or PPPoE headers.

5. Discovery Stage

There are four steps to the Discovery stage. When it completes, both peers know the PPPoE SESSION_ID and the peer’s Ethernet address, which together define the PPPoE session uniquely. The steps consist of the Host broadcasting an Initiation packet, one or more Access Concentrators sending Offer packets, the Host sending a unicast Session Request packet and the selected Access Concentrator sending a Confirmation packet. When the Host receives the Confirmation packet, it may proceed to the PPP Session Stage. When the Access Concentrator sends the Confirmation packet, it may proceed to the PPP Session Stage.

All Discovery Ethernet frames have the ETHER_TYPE field set to the value 0x8863.

The PPPoE payload contains zero or more TAGs. A TAG is a TLV (type-length-value) construct and is defined as follows:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          TAG_TYPE             |        TAG_LENGTH             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          TAG_VALUE ...                                        ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TAG_TYPE is a sixteen bit field in network byte order. Appendix A contains a list of all TAG_TYPEs and their TAG_VALUEs.

TAG_LENGTH is a sixteen bit field. It is an unsigned number in network byte order, indicating the length in octets of the TAG_VALUE.

If a discovery packet is received with a TAG of unknown TAG_TYPE, the TAG MUST be ignored unless otherwise specified in this document. This provides for backwards compatibility if/when new TAGs are added. If new mandatory TAGs are added, the version number will be incremented.

Some example Discovery packets are shown in Appendix B.

5.1 The PPPoE Active Discovery Initiation (PADI) packet

The Host sends the PADI packet with the DESTINATION_ADDR set to the broadcast address. The CODE field is set to 0x09 and the SESSION_ID MUST be set to 0x0000.

The PADI packet MUST contain exactly one TAG of TAG_TYPE Service-Name, indicating the service the Host is requesting, and any number of other TAG types. An entire PADI packet (including the PPPoE header) MUST NOT exceed 1484 octets so as to leave sufficient room for a relay agent to add a Relay-Session-Id TAG.

5.2 The PPPoE Active Discovery Offer (PADO) packet

When the Access Concentrator receives a PADI that it can serve, it replies by sending a PADO packet. The DESTINATION_ADDR is the unicast address of the Host that sent the PADI. The CODE field is set to 0x07 and the SESSION_ID MUST be set to 0x0000.

The PADO packet MUST contain one AC-Name TAG containing the Access Concentrator's name, a Service-Name TAG identical to the one in the PADI, and any number of other Service-Name TAGs indicating other services that the Access Concentrator offers. If the Access Concentrator can not serve the PADI it MUST NOT respond with a PADO.

5.3 The PPPoE Active Discovery Request (PADR) packet

Since the PADI was broadcast, the Host may receive more than one PADO. The Host looks through the PADO packets it receives and chooses one. The choice can be based on the AC-Name or the Services offered. The Host then sends one PADR packet to the Access Concentrator that it has chosen. The DESTINATION_ADDR field is set to the unicast Ethernet address of the Access Concentrator that sent the PADO. The CODE field is set to 0x19 and the SESSION_ID MUST be set to 0x0000.

The PADR packet MUST contain exactly one TAG of TAG_TYPE Service-Name, indicating the service the Host is requesting, and any number of other TAG types.

5.4 The PPPoE Active Discovery Session-confirmation (PADS) packet

When the Access Concentrator receives a PADR packet, it prepares to begin a PPP session. It generates a unique SESSION_ID for the PPPoE session and replies to the Host with a PADS packet. The DESTINATION_ADDR field is the unicast Ethernet address of the Host that sent the PADR. The CODE field is set to 0x65 and the SESSION_ID MUST be set to the unique value generated for this PPPoE session.

The PADS packet contains exactly one TAG of TAG_TYPE Service-Name, indicating the service under which Access Concentrator has accepted the PPPoE session, and any number of other TAG types.

If the Access Concentrator does not like the Service-Name in the PADR, then it MUST reply with a PADS containing a TAG of TAG_TYPE Service-Name-Error (and any number of other TAG types). In this case the SESSION_ID MUST be set to 0x0000.

5.5 The PPPoE Active Discovery Terminate (PADT) packet

This packet may be sent anytime after a session is established to indicate that a PPPoE session has been terminated. It may be sent by either the Host or the Access Concentrator. The DESTINATION_ADDR field is a unicast Ethernet address, the CODE field is set to 0xa7 and the SESSION_ID MUST be set to indicate which session is to be terminated. No TAGs are required.

When a PADT is received, no further PPP traffic is allowed to be sent using that session. Even normal PPP termination packets MUST NOT be sent after sending or receiving a PADT. A PPP peer SHOULD use the PPP protocol itself to bring down a PPPoE session, but the PADT MAY be used when PPP can not be used.

6. PPP Session Stage

Once the PPPoE session begins, PPP data is sent as in any other PPP encapsulation. All Ethernet packets are unicast. The ETHER_TYPE field is set to 0x8864. The PPPoE CODE MUST be set to 0x00. The SESSION_ID MUST NOT change for that PPPoE session and MUST be the value assigned in the Discovery stage. The PPPoE payload contains a PPP frame. The frame begins with the PPP Protocol-ID.

An example packet is shown in Appendix B.

7. LCP Considerations

The Magic Number LCP configuration option is RECOMMENDED, and the Protocol Field Compression (PFC) option is NOT RECOMMENDED. An implementation MUST NOT request any of the following options, and MUST reject a request for such an option:

Field Check Sequence (FCS) Alternatives,

Address-and-Control-Field-Compression (ACFC),

Asynchronous-Control-Character-Map (ACCM)

The Maximum-Receive-Unit (MRU) option MUST NOT be negotiated to a larger size than 1492. Since Ethernet has a maximum payload size of 1500 octets, the PPPoE header is 6 octets and the PPP Protocol ID is 2 octets, the PPP MTU MUST NOT be greater than 1492.

It is RECOMMENDED that the Access Concentrator ocassionally send Echo-Request packets to the Host to determine the state of the session. Otherwise, if the Host terminates a session without sending a Terminate-Request packet, the Access Concentrator will not be able to determine that the session has gone away.

When LCP terminates, the Host and Access concentrator MUST stop using that PPPoE session. If the Host wishes to start another PPP session, it MUST return to the PPPoE Discovery stage.

8. Other Considerations

When a host does not receive a PADO packet within a specified amount of time, it SHOULD resend it's PADI packet and double the waiting period. This is repeated as many times as desired. If the Host is waiting to receive a PADS packet, a similar timeout mechanism SHOULD be used, with the Host re-sending the PADR. After a specified number of retries, the Host SHOULD then resend a PADI packet.

The ETHER_TYPEs used in this document (0x8863 and 0x8864) have been assigned by the IEEE for use by PPP Over Ethernet (PPPoE). Use of these values and the PPPoE VER (version) field uniquely identify this protocol.

UTF-8 [5] is used throughout this document instead of ASCII. UTF-8 supports the entire ASCII character set while allowing for international character sets as well. See [5] for more details.

9. Security Considerations

To help protect against Denial of Service (DOS) attacks, the Access Concentrator can employ the AC-Cookie TAG. The Access Concentrator SHOULD be able to uniquely regenerate the TAG_VALUE based on the PADR SOURCE_ADDR. Using this, the Access Concentrator can ensure that the PADI SOURCE_ADDR is indeed reachable and can then limit concurrent sessions for that address. What algorithm to use is not defined and left as an implementation detail. An example is HMAC [3] over the Host MAC address using a key known only to the Access > Concentrator. While the AC-Cookie is useful against some DOS attacks, it can not protect against all DOS attacks and an Access Concentrator MAY employ other means to protect resources.

While the AC-Cookie is useful against some DOS attacks, it can not protect against all DOS attacks and an Access Concentrator MAY employ other means to protect resources.

Many Access Concentrators will not wish to offer information regarding what services they offer to an unauthenticated entity. In that case the Access Concentrator should employ one of two policies. It SHOULD never refuse a request based on the Service-Name TAG, and always return the TAG_VALUE that was sent to it. Or it SHOULD only accept requests with a Service-Name TAG with a zero TAG_LENGTH (indicating any service). The former solution is RECOMMENDED.

10. Acknowledgments

This document is based on concepts discussed in several forums, including the ADSL forum.

Copious amounts of text have been stolen from RFC 1661, RFC 1662 and RFC 2364.

11. References

[1] Simpson, W., Editor, "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[3] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1998.

[4] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994. See also: http://www.iana.org/numbers.html

[5] Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 2279, January 1998.

Appendix A

TAG_TYPES and TAG_VALUES

0x0000 End-Of-List

This TAG indicates that there are no further TAGs in the list. The TAG_LENGTH of this TAG MUST always be zero. Use of this TAG is not required, but remains for backwards compatibility.

0x0101 Service-Name

This TAG indicates that a service name follows. The TAG_VALUE is an UTF-8 string that is NOT NULL terminated. When the TAG_LENGTH is zero this TAG is used to indicate that any service is acceptable. Examples of the use of the Service-Name TAG are to indicate an ISP name or a class or quality of service.

0x0102 AC-Name

This TAG indicates that a string follows which uniquely identifies this particular Access Concentrator unit from all others. It may be a combination of trademark, model, and serial id information, or simply an UTF-8 rendition of the MAC address of the box. It is not NULL terminated.

0x0103 Host-Uniq

This TAG is used by a Host to uniquely associate an Access Concentrator response (PADO or PADS) to a particular Host request (PADI or PADR). The TAG_VALUE is binary data of any value and length that the Host chooses. It is not interpreted by the Access Concentrator. The Host MAY include a Host-Uniq TAG in a PADI or PADR. If the Access Concentrator receives this TAG, it MUST include the TAG unmodified in the associated PADO or PADS response.

0x0104 AC-Cookie

This TAG is used by the Access Concentrator to aid in protecting against denial of service attacks (see the Security Considerations section for an explanation of how this works). The Access Concentrator MAY include this TAG in a PADO packet. If a Host receives this TAG, it MUST return the TAG unmodified in the following PADR. The TAG_VALUE is binary data of any value and length and is not interpreted by the Host.

0x0105 Vendor-Specific

This TAG is used to pass vendor proprietary information. The first four octets of the TAG_VALUE contain the vendor id and the remainder is unspecified. The high-order octet of the vendor id is 0 and the low-order 3 octets are the SMI Network Management Private Enterprise Code of the Vendor in network byte order, as defined in the Assigned Numbers RFC [4].

Use of this TAG is NOT RECOMMENDED. To ensure inter-operability, an implementation MAY silently ignore a Vendor-Specific TAG.

0x0110 Relay-Session-Id

This TAG MAY be added to any discovery packet by an intermediate agent that is relaying traffic. The TAG_VALUE is opaque to both the Host and the Access Concentrator. If either the Host or Access Concentrator receives this TAG they MUST include it unmodified in any discovery packet they send as a response. All PADI packets MUST guarantee sufficient room for the addition of a Relay-Session-Id TAG with a TAG_VALUE length of 12 octets.

A Relay-Session-Id TAG MUST NOT be added if the discovery packet already contains one. In that case the intermediate agent SHOULD use the existing Relay-Session-Id TAG. If it can not use the existing TAG or there is insufficient room to add a Relay-Session-Id TAG, then it SHOULD return a Generic-Error TAG to the sender.

0x0201 Service-Name-Error

This TAG (typically with a zero-length data section) indicates that for one reason or another, the requested Service-Name request could not be honored.

If there is data, and the first octet of the data is nonzero, then it MUST be a printable UTF-8 string which explains why the request was denied. This string MAY NOT be NULL terminated.

0x0202 AC-System-Error

This TAG indicates that the Access Concentrator experienced some error in performing the Host request. (For example insufficient resources to create a virtual circuit.) It MAY be included in PADS packets.

If there is data, and the first octet of the data is nonzero, then it MUST be a printable UTF-8 string which explains the nature of the error. This string MAY NOT be NULL terminated.

0x0203 Generic-Error

This TAG indicates an error. It can be added to PADO, PADR or PADS packets when an unrecoverable error occurs and no other error TAG is appropriate. If there is data then it MUST be an UTF-8 string which explains the nature of the error. This string MUST NOT be NULL terminated.

Appendix B

The following are some example packets:

A PADI packet:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         0xffffffff                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           0xffff              |        Host_mac_addr          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Host_mac_addr (cont)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    ETHER_TYPE = 0x8863        | v = 1 | t = 1 |  CODE = 0x09  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     SESSION_ID = 0x0000       |      LENGTH = 0x0004          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      TAG_TYPE = 0x0101        |    TAG_LENGTH = 0x0000        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A PADO packet:

```
                         1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         Host_mac_addr                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      Host_mac_addr (cont)     | Access_Concentrator_mac_addr  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Access_Concentrator_mac_addr (cont)               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    ETHER_TYPE = 0x8863        | v = 1 | t = 1 |  CODE = 0x07  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     SESSION_ID = 0x0000       |       LENGTH = 0x0020         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      TAG_TYPE = 0x0101        |    TAG_LENGTH = 0x0000        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      TAG_TYPE = 0x0102        |    TAG_LENGTH = 0x0018        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     0x47      |     0x6f      |     0x20      |     0x52      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     0x65      |     0x64      |     0x42      |     0x61      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     0x63      |     0x6b      |     0x20      |     0x2d      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     0x20      |     0x65      |     0x73      |     0x68      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     0x73      |     0x68      |     0x65      |     0x73      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     0x68      |     0x6f      |     0x6f      |     0x74      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A PPP LCP packet: The PPP protocol value is shown (0xc021) but the PPP payload is left to the reader. This is a packet from the Host to the Access Concentrator.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Access_Concentrator_mac_addr                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Access_Concentrator_mac_addr(c)|        Host_mac_addr          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Host_mac_addr (cont)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    ETHER_TYPE = 0x8864        | v = 1 | t = 1 |  CODE = 0x00  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     SESSION_ID = 0x1234       |      LENGTH = 0x????          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    PPP PROTOCOL = 0xc021      |        PPP payload            ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Authors' Addresses

Louis Mamakos
UUNET Technologies, Inc.
3060 Williams Drive
Fairfax, VA 22031-4648
United States of America

EMail: louie@uu.net

Kurt Lidl
UUNET Technologies, Inc.
3060 Williams Drive
Fairfax, VA 22031-4648
United States of America

EMail: lidl@uu.net

Jeff Evarts
UUNET Technologies, Inc.
3060 Williams Drive
Fairfax, VA 22031-4648
United States of America

EMail: jde@uu.net

David Carrel
RedBack Networks, Inc.
1389 Moffett Park Drive
Sunnyvale, CA 94089-1134
United States of America

EMail: carrel@RedBack.net

Dan Simone
RedBack Networks, Inc.
1389 Moffett Park Drive
Sunnyvale, CA 94089-1134
United States of America

EMail:dan@RedBack.net

Ross Wheeler
RouterWare, Inc.
3961 MacArthur Blvd., Suite 212
Newport Beach, CA 92660
United States of America

EMail: ross@routerware.com

Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.